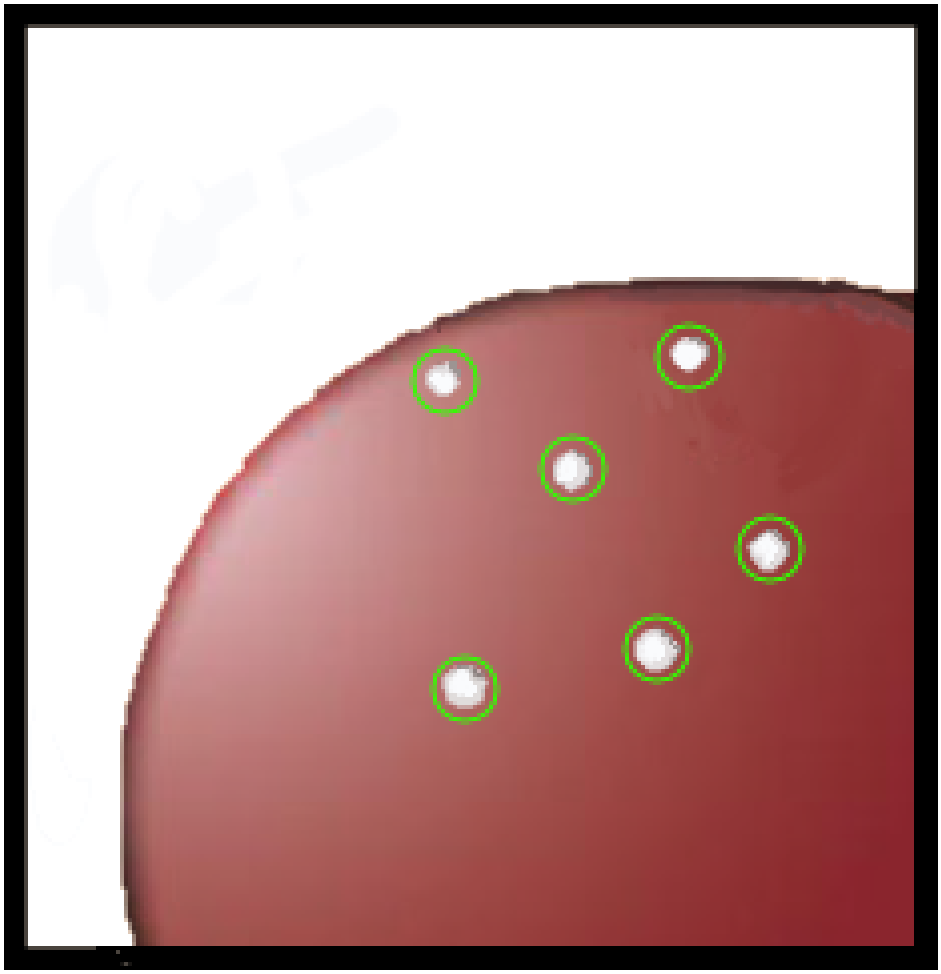
CHALMERS
UNIVERSITY OF TECHNOLOGY



A Model for Object Recognition in Liver Resection Surgery

Master's thesis in Engineering Mathematics and Computational Science

CHRISTIAN AL-MALEH

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020

A Model for Object Recognition in Liver Resection Surgery

CHRISTIAN AL-MALEH

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Mathematical Statistics
Chalmers University of Technology
Gothenburg, Sweden 2020

A Model for Object Recognition in Liver Resection Surgery
CHRISTIAN AL-MALEH

Supervisor: Torbjörn Lundh, Department of Mathematical Sciences
Examiner: Torbjörn Lundh, Department of Mathematical Sciences

Master's Thesis 2020
Department of Mathematical Sciences
Division of Applied Mathematics and Mathematical Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An illustration of landmarks being detected in a two-dimensional view.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

A Model for Object Recognition in Liver Resection Surgery
CHRISTIAN AL-MALEH
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Laparoscopic liver resection is a safer alternative to open surgery for treatment of liver cancer, a disease which claims almost 800 000 lives every year. The procedure involves making small incisions in the abdomen where instruments and a camera, called a laparoscope, are inserted. One of the major drawbacks of laparoscopic surgery is the restricted view and orientation, as well as lack of haptic feedback. Incorporating Augmented Reality, or AR, in the laparoscopic view is a proposed method of facilitating the navigation. This work extends a previous model for projecting information from 3D to 2D and vice versa using reference points, which correctly visualizes the shape, angle and size of a tumor in AR in the 2D laparoscopic view. To enable the 2D-to-3D projection, two object recognition models based on image segmentation and edge detection, respectively, were developed where white reference objects were distinguished from the darker tones of the liver tissue. The positions of the reference objects were then measured. The latter model, albeit effective given certain frames, failed to identify fiducials over the course of a test film. Since the process of image segmentation is computationally heavy, it was localized to an area of interest in a given frame, reducing the algorithm's runtime. Statistical error estimation was used to validate the positions found by this recognition model. The average position error produced was between 1 to 5 pixels, where the frames had a pixel height of 1080. Future work involves combining the recognition algorithm with the projection model to examine the effect of the deviations of the estimated positions in the 2D laparoscopic view.

Keywords: object recognition, image segmentation, edge detection, laparscopic, liver cancer, surgery, augmented reality.

Acknowledgements

Firstly, I would like to thank my supervisor Torbjörn Lundh for his vision, directions and advice when working on this master's thesis. I would also like to thank Lisa Månsson and Klas Modin, whose invaluable input lead to a nuanced understanding of the problem at hand.

I would like to thank Niclas Kvarnström and Mårten Falkenberg for their collaboration and guidance. I appreciate your endless hospitality and my tours around the liver surgery unit at Sahlgrenska University Hospital have been a great experience.

The meetings with all five of you have been nothing short of insightful and fun.

Finally, I would like to thank my friends and family for their inexhaustible support throughout the course of this project.

Christian Almaleh, Gothenburg, June 2020

Contents

List of Figures

1
Introduction

The development of laparoscopic liver resection, or LLR, has introduced a safer treatment for liver cancer, as this technique for tumor removal enables for faster postoperative recovery and lower risk for infections, which are otherwise common complications of open surgery where larger incisions are made [1, 2]. As such, liver resection is remains one of the most viable options for this type of cancer, which claims 780 000 lives annually and has the fourth most common cancer deaths [3]. LLR is a form of minimally invasive surgery, where small incisions are made and into which surgical tools and a camera, or laparoscope, are inserted, after the abdomen has been inflated with carbon dioxide in a procedure called *pneumoperitoneum* [4]. Through the laparoscope, surgeons can navigate the abdomen and perform the resection as the footage captured by the laparoscope is viewed on a monitor beside the operating team.

A quicker recovery means fewer adverse effects in terms of scarring, blood loss, abdominal pain, and impediment in physical movement, allowing patients to return to normal activity sooner, with fewer hospital visits in the future [5].

Despite its benefits, LLR has crucial drawbacks. It is mostly limited to peripheral resections, meaning tumors embedded in the liver, which has a quite homogeneous surface, are more difficult to resect in laparoscopic surgery [6]. To facilitate an easier navigation during such cases, computer topography scans (CT) and magnetic resonance imaging (MRI) of the liver can help guide the surgeons. Nevertheless, it ultimately falls on the surgeon to resect most of the malignant tumor, while simultaneously avoiding as much healthy liver tissue, which remains a problem due to the restricted view.

To this end, various methods involving Augmented Reality, or AR, have been researched and implemented to facilitate the laparoscopic orientation. Below follows two particular examples, one shares a similar premise with this thesis while the other serves as a building ground for it.

1.1 Previous work

Augmented Reality is, simply put, supplementary information added to images or video [7]. Organizing this process in a way where tracking points of interest dictates the location and shape of an AR object is possible through means of image analysis,

computer vision and optical tracking. Through algorithms of these fields, features of video frames or images are extracted [8], which may include colors or edges of objects. Given this knowledge, properties such as an objects center position may be determined.

This concept is relevant for laparoscopic surgery, where video frames capturing a resection live can be processed in similar way. In one instance, optical tracking references were attached to the certain instruments and to the camera, and through video calibration and patient-to-image registration, preoperative[1] 3D information on the liver was displayed in 2D as AR [9].

1.1.1 Fiducial-based tracking

Similar to the example with optical tracking above, inserting or marking reference objects directly on to the liver to serve as tracking points in 2D has been implemented by Teatini et al. (2019) [4]. These landmarks, or *fiducials*, are registered in the 2D view through an optical tracking device attached to the laparoscope, as in the previous example. The tracking procedure follows and algorithm in computer vision called *hand-eye calibration*, to find a transformation matrix between an instruments tracking sources and the laparoscopic pose. When this is done, an *image-to-patient registration* is performed, which involves transforming the content of a CT scan to the patient's position and orientation on the surgical table.

As this procedure was performed *in vivo*[2], the effects of the abdominal inflation deformed the soft tissue of the live, which decreases the accuracy of the transformations computed and in turn the AR projection. That is, despite correctly identifying the fiducials, these position do not necessarily align to those in a preoperative image scan. To rectify these skewed misplacements, image scanning is done intraoperatively[3].

1.1.2 Augmented Reality tool

As mentioned, this thesis directly follows the work by Lisa Månsson and her master thesis (2019) [10]. In it, two models are constructed and combined. The first one, called the *forward camera model*, simulates a liver environment with a tumor in 3D and, given the known points of fiducials in the figure, projects them down to a plane in 2D using perspective projection, serving as a camera view. The solution to the inverse problem is the second model, where having true or estimated points in the 2D plane, one projects the information to 3D. For this problem, an algorithm called POSIT is utilized to estimate the camera pose in relation to an object in 3D using a frame or image containing a number of fiducials, which also have corresponding in the 3D figure. The result of this is an algorithm that tracks positions in a simulated 2D camera view, and given the tumor's position in the emulated liver in 3D, returns

[1] Preparations occurring before surgery.
[2] Performing an experiment inside the patient.
[3] In this context: scanning the abdomen during surgery.

the corresponding tumor as perceived from that 2D position.

In contrast to the previous model, the Augmented Reality tool does not need additional equipment for transforming between 3D to 2D, and vice versa. However, a step that was not developed was the tracking of fiducials in 2D. Moreover, since the environment in which the algorithm was implemented was simulated, certain noise factors associated from captured footage, such as brightness, were not accounted for.

One obstacle particular to POSIT is that in some camera orientation, the output experiences spikes of error. This is due to the fact that two of the algorithm's assumptions. Firstly, that the distance between the camera and the fiducials must be larger than that of between fiducials. Secondly, the fiducials have to be non-coplanar[4]

1.2 Aim

The reprojection model constructed by Månsson can be supplemented where an input for the transformation, namely the fiducial positions in the 2D laparoscopic view, is defined. To follow suit with the idea of avoiding the use of additional equipment in the operation room, as was the case for the Augmented Reality tool, this thesis is focused on designing an algorithm for tracking the fiducials in laparoscopic frames strictly using principles of image analysis, more specifically object recognition. Object recognition has been used for a variety of tasks, such as face detection [11], traffic sign identification [12] and activity recognition [13], to name a few.

As mentioned before, because the fiducials in 2D in the reprojection model are simulated, crucial aspects of noise, such as lighting-induced noise, and irregularities caused by the patient's condition, such as cirrhosis[5], are absent. To materialize the solution presented by Augmented Reality tool, an object recognition algorithm centered around distinguishing fiducials placed on actual liver tissue, as well as estimating the fiducial positions in 2D, needs to be designed. For the recognition model to be feasible it is required to perform its task automatically, without user interference, and generally without interruptions between frames. Also, although errors are to be expected when determining the position of each respective fiducial, finding ways to increase accuracy is a priority, whether it relates to the functionality of the algorithm or the setup for the test environment, such as selecting the color or shape of the fiducials. It is important to note that even a slightly erroneous input may possibly affect the transformation from 3D to 2D quite drastically, and as such yield a misleading and incorrect AR projection. Lastly, knowing about the limitations of the POSIT algorithm, conforming to its assumptions is key.

In this thesis, two recognition algorithms are studied. One separates the fiducials

[4]Objects that do not lie on the same surface or plane.
[5]A liver condition characterized by scarring tissue.

from the background based on their difference in brightness. The other locates the edges if the fiducials in order to differentiate them. The two recognition algorithms are compared further at the start of chapter 3.

Due to hospital regulations, tests can only be performed *ex vivo*[6]. Under the supervision of surgeons Niclas Kvarnström and Mårten Falkenberg at Sahlgrenska University hospital, the activity of a typical laparoscopy is recreated in shorter films. For this thesis, the conclusions of the experiment and its results are thus predicated on laparoscopic footage of a liver specimen, ex vivo. Consequently, the element of potential body movement affecting the liver position, and possibly the fiducials, will be disregarded. Moreover, the ex vivo environment does not account for surrounding abdominal tissue which will affect certain parameter values and settings, as detailed further when presenting the method of the thesis. Nevertheless, the general principle of the methods selected and algorithm constructed remains constant. Lastly, although the recognition techniques can largely be translated to in vivo, the tissue deformation caused by inflating the abdomen with carbon dioxide preoperatively is absent in the ex vivo model.

1.2.1 Issue specification

The three specific assignments at hand for this thesis are as follows:

1. To compare the two methods of edge detection and image segmentation in finding the fiducials.
2. To design an algorithm which can identify the fiducials and estimate their center position, whilst conforming to the POSIT assumptions and how the surgeons operate.
3. Perform an error analysis on the fiducials' estimated positions that is sufficient in describing the performance of the identification.
4. Efficiently reduce the computation time of the algorithm, while retaining similar quality to that of the original laparoscopic display.

To illustrate this thesis in relation to the previous work of the Augmented Reality tool, a work-flow chart is presented in figure 1.1

[6]Performing a laboratory experiment, as opposed to in vivo.

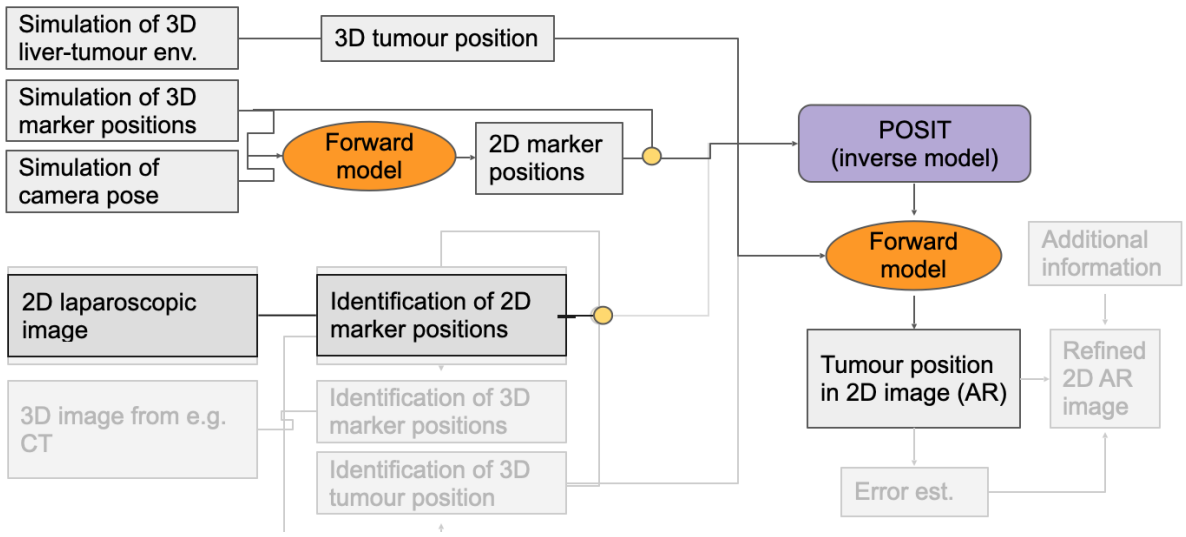


Figure 1.1: The flow chart documents the complete framework of the laparoscopic AR model. The steps concerning the projection, most notably the POSIT algorithm, have been constructed beforehand. The identification of markers, or fiducials, is central in this thesis.

2

Theory

In this chapter, the theory behind the parts of the fiducial recognition algorithm is presented. Steps of how video frames are defined and how they are manipulated through grayscale conversion are first described. Then, the basics of image segmentation and edge detection are detailed. These two concepts are the focal points of the method of this thesis, as they provide the means to highlight the fiducials in simpler forms, or more specifically *binary images*. Lastly, utilities related to estimating the center position of regions in a binary image are defined.

2.1 Image properties

Before proceeding with the theory behind the methods utilized, a definition for video frames is required. Along with this, the component in which features are evaluated for the image segmentation, namely *grayscale* space, is defined.

2.1.1 Defining video frames

Simply put, a video frame can be regarded as a $m \times n \times 3$ matrix, where m, $n \in \mathbb{Z}$ and each element of the matrix corresponds to a pixel in the video frame. The first two dimensions represents the height and width of the video frame. The last dimension refers to the color information. The latter consists of three components, namely a red, green, and blue component. The intensity range for each component is [0, 255].

2.1.2 Grayscale image

In a *grayscale* image, the pixel intensities correspond to levels of brightness [14]. Removing the color information, or *chrominance* component from a colored image, while preserving the *luminance* component, yields the grayscale version of the image. Images in the RGB color space do not have luminance component defined separately, presenting an obstacle when seeking for the amount of brightness in the images. A solution to this is to transform the images to a color space with a distinct luminance component. The National Television Systems Committee, or NTSC, defines a color space called YIC, with the component Y carrying the luminance information [15]. The conversion to YIC involves multiplying the RGB channels with a transformation

matrix, as per following formula

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The pixel values of the grayscale image is thereby determined through the weighted sum

$$Y = 0.229 \cdot R + 0.587 \cdot G + 0.114 \cdot B \tag{2.1}$$

2.2 Image segmentation

One of the solutions for fiducial recognition is based on image segmentation, where areas of interest are highlighted based on certain conditions. This results in a binary image, where the pixels are grouped into two colors, such as white and black. A formal definition of a binary image, along with ways to utilize it, follows in the sections below.

2.2.1 Binary image

As the name implies, a binary image is comprised of two colors. Where, for example, pixels fulfilling certain conditions assume one color, while remaining pixels assume the other color. Let an arbitrary set of constraints be defined as C. The pixels of a $m \times n$, for some m, $n \in \mathbb{Z}$, binary image B are defined as

$$B_{ij} = \begin{cases} 1, & c \in C, \text{ for some value } \ c \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

Here, $B_{ij} = 1$ is shown as white, while $B_{ij} = 0$ is shown as black.

A similarity to grayscale images can be drawn in that binary images only assume two values for brightness, namely the brightest intensity at value 1 and the darkest intensity at value 0.

Thesholding is a way to generate binary image, where pixels values, in for example a grayscale image, that are greater or less than a set threshold yields one of the colors in the binary image in the corresponding position.

2.2.2 Connected-component labeling

The collection of algorithms which distinguish regions in a binary image is called connected-component labeling [16, 17]. One of these algorithms, which is also applied for this thesis, is called flood-fill algorithm [18, 16]. It entails starting at a white pixel and counting its adjacent pixels, or neighbours, marking these under one label. This is repeated, with the same label as before, for each of the neighbours. When there are no more adjacent neighbours, the algorithm moves on to a pixel of another region, treating it under a different label. More formally, the steps

are

1. Let the binary image B be an $m \times n$ matrix, with pixels defined as $B_{ij} \in \{1, 0\}$ for $i = 1, ..., M$ and $j = 1, ..., N$. Let the *label matrix* L be an $m \times n$ zero matrix and $k \in \mathbb{N}$, which initially assumes value 1.

2. A white pixel, or $B_{ij} = 1$, in B is linearly searched for. If the value in the corresponding index of L is nonzero, then the search is continued until a white pixel whose corresponding location (i, j) in L is zero is found.

3. Set $L_{ij} = k$, which labels the pixel B_{ij} under the integer k.

4. The pixels adjacent to B_{ij}, or neighbours, are defined by a $d \times 2$ neighbour set $A := \{\forall k \in (i \pm 1, j \pm 1), (i, j \pm 1), (i \pm 1, j) | B_k = 1;\ i = 1, ..., n; j = 1, ..., m\}$. The size of a neighbourhood is thus at most 8 pixels, which is based on 8-connectivity. Here, $0 < d < 8$ refers to the number of indices k where $B_k = 1$.

5. For each neighbour $1 \leq r \leq d$ in A, where the index of r is given as (i_r, j_r), step (3) to (4) are repeated, before removing neighbour r from A. This in turn yields new neighbours, denoted by the set A', to the pixels in A. These pixels are similarly processed through step (3) to (4).

6. Once every pixel in A and corresponding A', are removed, let $k = k + 1$ and repeat steps (2) to (5).

7. Once every pixel in B is processed through steps (2) to (6), the regions are labeled.

1	1	0	0	0
1	1	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	1	1	0

1	1	0	0	0
1	1	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	2	2	0

Figure 2.1: The product of applying the flood-fill algorithm on a binary image (left) is the label matrix (right) where each disjoint and contiguous group of pixels with value 1 in the binary image is labeled with some integer.

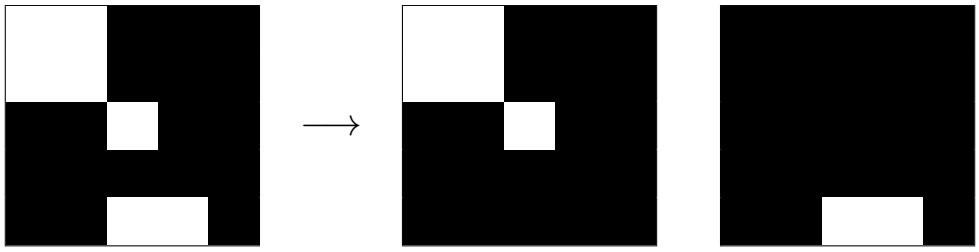


Figure 2.2: The results from 2.1 depicted as black-and-white binary images. On the right hand side, the original binary image B can subsequently be split into two versions $B^{(1)}$ (left) and $B^{(2)}$ (right) containing a designated region, as shown in the left hand side.

2.3 Edge detection

The other solution for identifying the fiducials involves finding its edges. From there, using morphological reconstruction to fill regions enclosed by the edges, thereby yielding a binary image with similar characteristics to that of image segmentation in section 2.2.

One way of evaluating edge strength[1] is by measuring the directional changes in, for example, color [19]. This is characterized by sharpness or color contrasts in images. For this task, the first-order image gradient G is approximated according to the convolutions

$$G_x = F_x * I \tag{2.3}$$
$$G_y = F_y * I \tag{2.4}$$

where I is a two-dimensional image matrix. The terms F_x and F_y are called convolutional kernels, or filters. Various forms of these kernels have been proposed. Notable ones include the Sobel filters, the Roberts cross filters, and the Prewitt filters [20, 21].

The Sobel filters are given as

$$F_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \qquad F_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2.5}$$

These kernels respond in particular to horizontal and vertical edges [22].

Similar to the Sobel filters, the Prewitt filters are formulated as

$$F_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, \qquad F_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.6}$$

[1] A measure of how well-defined a given edge is.

Despite the similarities between the Sobel and Prewitt filters, the latter is not as isotropic in response[2] to edges in a given image [22].

Lastly, the Roberts cross filters are instead comprised the 2×2 kernels,

$$F_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad F_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} \tag{2.7}$$

The Roberts cross filters primarily detect edges running diagonally to the pixel grid [23]. Due to its smaller kernels, the Roberts cross filter smooths the input image less compared to the Sobel and Prewitt filters. In turn, the approximation of the image gradient with the Roberts cross kernels is more susceptible to noise. On the other hand, the approximation using the larger Sobel or Prewitt kernels is slower to compute.

For the edge strength, the gradient magnitude is computed as

$$|\nabla G| = \sqrt{G_x^2 + G_y^2} \tag{2.8}$$

From this point, a binary image can be produced containing a set of edges specified by thresholding the gradient magnitude [21]. That is, for an arbitrary set of edge pixels E, a binary image B is defined as

$$B = \begin{cases} 1, \ |\nabla G_e| > t, \ e \in E \\ 0, \ \text{otherwise} \end{cases} \tag{2.9}$$

where a white pixel in B constitutes an edge pixel whose gradient magnitude $|\nabla G_e|$ is greater than a threshold $t > 0$.

2.3.1 Morphological reconstruction

In order to extract the complete regions of the fiducials, the area encapsulated by the detected edges, or holes, need to be filled. This process is called *morphological reconstruction*, and may implement procedures related to the flood-fill operation in section 2.2.2 [24].

Let B be a binary image containing an arbitrary amount of edges. The edge pixels comprise of the white pixels, while the remaining pixels in the background, including the holes, are black. The holes are then defined as a set of background pixels that cannot be filled if the algorithm starts from the edges of the frame [25].

In terms of applying the flood-fill algorithm, consider starting at a background pixel at some position on the image border, moving through each row of the image linearly just as described in section 2.1. Each background pixel's position is noted and labeled, using a label matrix. As before, the background pixels adjacent neighbours

[2] In this context, the Prewitt kernels may not detect edges in as many orientations as the Sobel kernels.

are labeled as well if they too are of value 0. After these neighbours have been noted, the labeling procedure repeats starting at their position, and so on. While neighbouring edge pixels do not serve as starting points for this procedure, their positions are registered in order to eventually distinguish holes from the remaining pixels. As opposed to how connected-components were labeled, the background pixels are labeled through 4-connectivity. This means that only adjacent pixels to the north, east, south and west are considered neighbours. With 8-connectivity, where pixels adjacent diagonally are neighbours, there is potential for crossing edges and mislabeling pixels in holes, which disrupts the result. This problem is illustrated in figure 2.3 below.

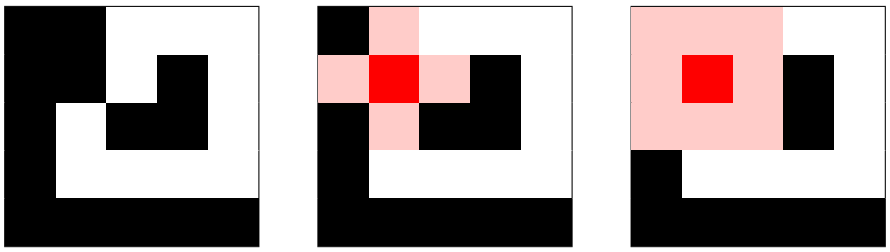


Figure 2.3: The effect of using 4-connectivity as neighbour structure as opposed to 8-connectivity. The leftmost figure depicts a hollow object, with its edge outlined. The center figure shows the reach of 4-connectivity, with the red pixel as starting position. As diagonal neighbours are not considered, the edge is not crossed and the pixels in the holes are not affected. Since edge pixels are not accounted for, they are simply registered and the algorithm moves on to next pixel candidate. The rightmost figure instead portrays the issue with 8-connectivity, where diagonal neighbours, and in turn pixels in the hole, are incorrectly labeled.

3

Methods

The identification of various shapes and colors in images is an established concept in image analysis. For this case, detecting circles and estimating their positions can be achieved using processes such as image segmentation and edge detection. In this project, both these methods will be explored. In this section, the steps taken to implement the methods in order to achieve the thesis' goals are detailed. To begin, the instruments used in the experiment as well as a description of the data sample are briefly reviewed. Secondly, the image segmentation process is presented and the way it is specifically used for collectively distinguishing the fiducials in a video frame. Next, the edge detection process is described in a similar manner. To validate the results, an error analysis is performed. Lastly, directions taken towards optimizing the algorithm in terms of runtime, will be presented and discussed.

All code used to solve the stated problem was written in MATLAB 2019b.

3.1 Equipment and data

The video was taken using a laparoscope with a camera light, manufactured by the company Olympus. The focal length of the laparoscope was 1600 pixels. The dimensions of the video frames are 1080x1918x3 where the last dimension refers to the channels in the RGB color space. As the video frames are treated as matrices, where each pixel corresponds to a matrix element, the coordinates of the frames are given in pixels. Furthermore, the values on the horizontal and vertical axes, or x and y coordinates, are equivalent to the elements on the columns and the rows of the matrix, respectively. An frame from a test film which will be covered in this project is presented in figure 4 below.

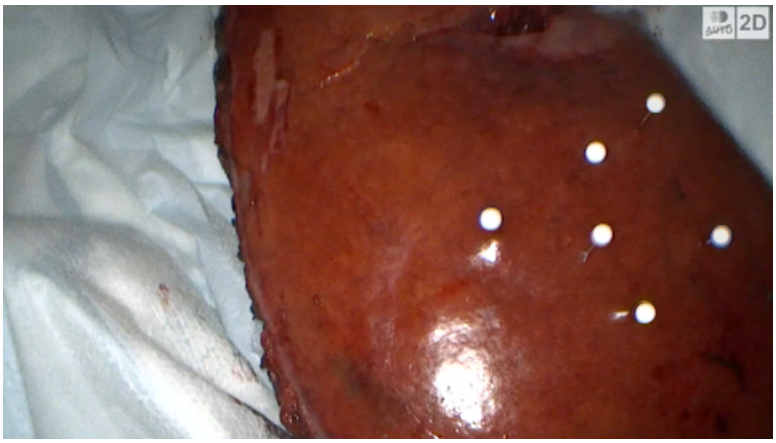


Figure 3.1: A frame from a laparoscopic test film. Six white fiducials were inserted on a liver specimen, which is placed on a white paper towel.

The ex vivo test environment included a specimen of a human liver on a bed of material, such as a sheet of white paper towel, in sculpted abdomen with incision ports. The surface of the sculpted abdomen consisted of white plastic. Lastly, regular pinhead needles with spherical heads of three millimeter diameter were used as fiducials. The selection of colors for the fiducial heads is crucial, as they must be easily separable from the red tones of the liver in terms of brightness. The color of the liver is quite dark in comparison to the color white in grayscale. Therefore, white fiducial heads are elected. As far as detecting edges goes, the stark color contrast should in theory allow for a more robust detection. Also, the notion that the liver tissue is fairly homogeneous suggests that an edge detection method is a sensible approach.

3.2 Comparing edge detection to brightness segmentation

A central aspect of this thesis which will be treated in this section, as well as in chapter 4 and 5, is the benefits and drawbacks of utilizing edge detection and brightness segmentation as tools for identifying the fiducials.

The more fundamental parts of the hypothesis is the amount of noise present in the resulting binary images after using these two methods. The general notion is that since edges tend to be slim, the noise regions after edge detection contain fewer pixels. This naturally results in noise regions of smaller area compared to reconstructed fiducials and are thereby easier to remove, as will be shown in section 3.5. As for the image segmentation, larger patches can be bright in the grayscale image and in turn appear in the binary image. The removal of these noise blobs may become complicated if their size is equal to that of the fiducials. However, as for computation time, image segmentation is more favorable.

3.3 Fiducial recognition - image segmentation

This section details the steps of extracting the fiducials in a given frame through the use of image segmentation. The focus will be in explaining the process of binary thresholding, as briefly mentioned in section 2.2.1, where regions in an image will be segmented if they surpass a set threshold based on brightness in the grayscale counterpart.

3.3.1 Extracting regions of interest

Applying formula in equation 3.2 on every pixel in a given frame, the resulting grayscale image can be defined as the 1080×1918-dimensional matrix G.

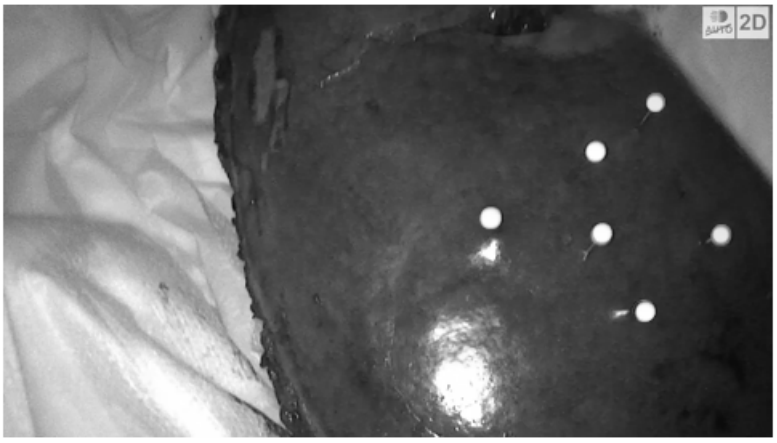


Figure 3.2: The grayscale-transformed frame in figure 4. The stark contrast between the white fiducials and the remaining tissue allows for a well-defined threshold when segmenting these objects from the background, where pixels with a brightness below a given intensity are omitted. The reflected light on the bottom-left corner of the liver, however, can exceed this threshold as noise.

To separate the fiducials from the liver tissue and remaining surroundings, a binary image is created with the following condition, as per the expression in equation 2.2,

$$B_{ij} = \begin{cases} 1, & G_{ij} > g \\ 0, & \text{otherwise} \end{cases}$$

That is, if the brightness exceeds a value g for a pixel in G the position in the binary image B assumes value 1, and the color white. If this condition is not fulfilled, the pixel values in B are 0, and thus black. The amount of noise, or unwanted regions, in the binary image correlates to the amount of separation between the brightness of

the fiducials and the remaining image in grayscale. In order to segment the entirety of the fiducials, a certain amount of noise may therefore appear. The resulting binary image is visualized in figure 3.3.

Figure 3.3: The resulting binary image after setting the brightness threshold at $G_{ij} > 220$. Note that the threshold may vary and is not necessarily universal, a topic which will be discussed in section 5. The fiducials are separated from the background, but a level of noise is present as well. The aforementioned reflected light in figure 3.2 is captured, for example.

3.4 Fiducial recognition - edge detection

This part walks through the steps of identifying the fiducial regions using edge detecting techniques as described in section 2.3. The three edge detection kernels, namely Sobel, Roberts cross and Prewitt filters, will be explored. Finally, the morphological reconstruction to fill the outlined fiducials detected is implemented.

3.4.1 Extracting edges of interest

After converting a video frame to grayscale as in section 3.3.1, the image gradient is computed according to the expressions in equation 2.3 and 2.4. To maintain low noise levels while ensuring that the fiducial edges come through, various threshold levels for the gradient magnitude are tested. Along with this, the Sobel, the Roberts cross, and the Prewitt kernels are compared.

Applying the edge detection with the Sobel filter for threshold of 0.005, 0.05 and 0.1 yields binary images as shown in figures 3.4, 3.5 and 3.6.

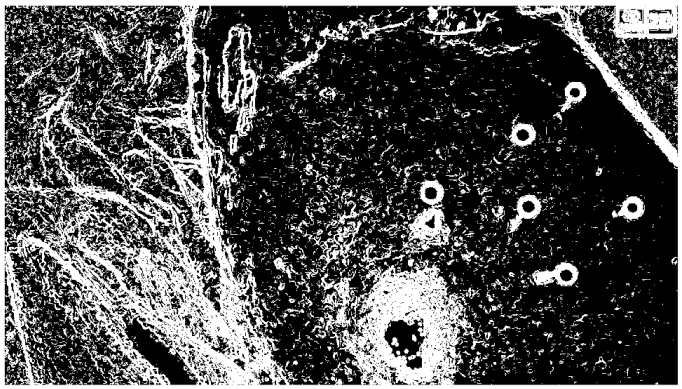


Figure 3.4: Detected edges using the Sobel kernels with a threshold of 0.005. While the fiducials are very accentuated, so are the needles on which the spherical heads are attached. Moreover, minuscule edges present all over the frame for this threshold making noise a significant factor.

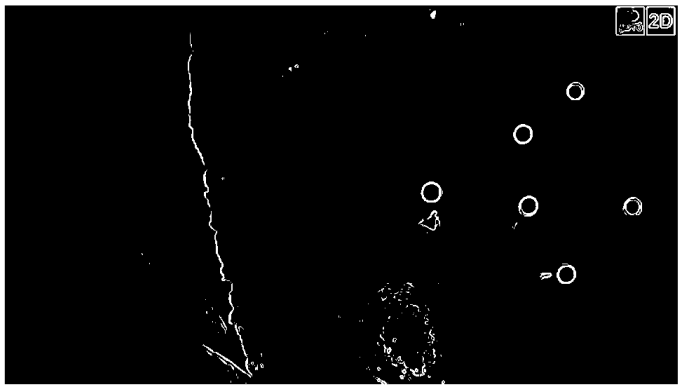


Figure 3.5: Detected edges using the Sobel kernels with a threshold of 0.05. The noise is largely reduced while the fiducial heads are sufficiently pronounced. The scattering of noise particles makes for an easier noise filtering at a later stage.

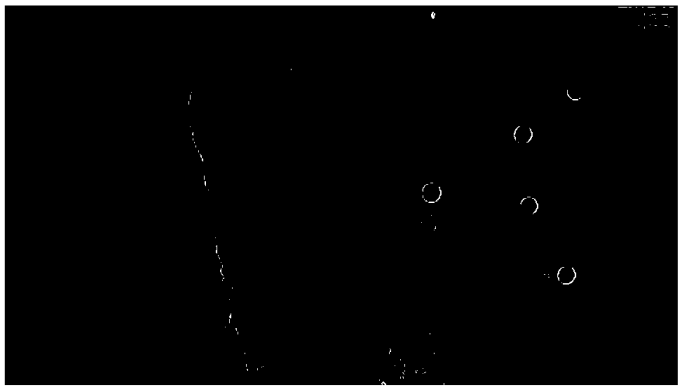


Figure 3.6: Detected edges using the Sobel kernels with a threshold of 0.1. This setting yields an inadequate amount of fiducial edges to be feasible.

With the same thresholds, the Roberts cross counterpart is shown in figures 3.7, 3.8 and 3.9.

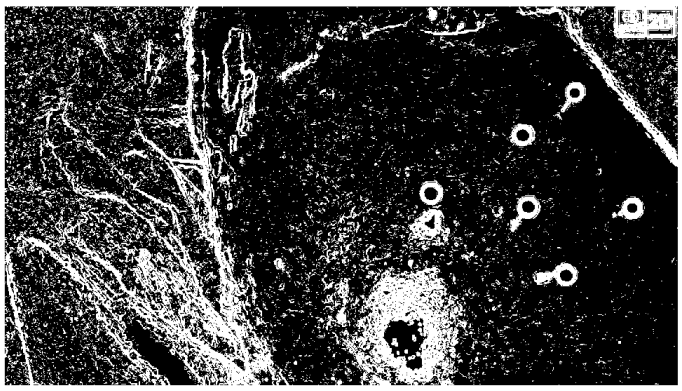


Figure 3.7: Detected edges using the Roberts cross kernels with a threshold of 0.005. The product is similar to when Sobel kernels are used.

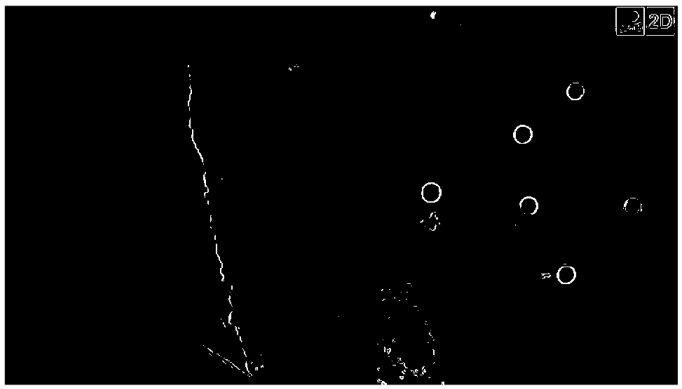


Figure 3.8: Detected edges using the Roberts cross kernels with a threshold of 0.05. While most noise is discarded, the fiducial edges are not properly presented.

Figure 3.9: Detected edges using the Roberts cross kernels with a threshold of 0.1. At this point, only a few particle of the fiducial edges and noise remain.

Lastly, the results of using the Prewitt filter analogously are presented in figures 3.10, 3.11 and 3.12.

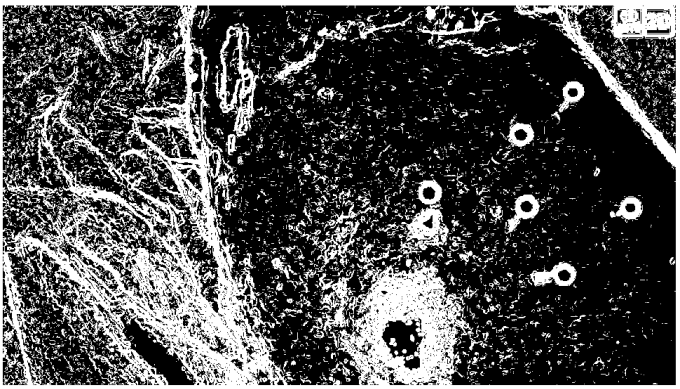


Figure 3.10: Detected edges using the Prewitt kernels with a threshold of 0.005. As with the previous two detectors, an abundance of noise is present while the edges of the fiducials are well-defined.

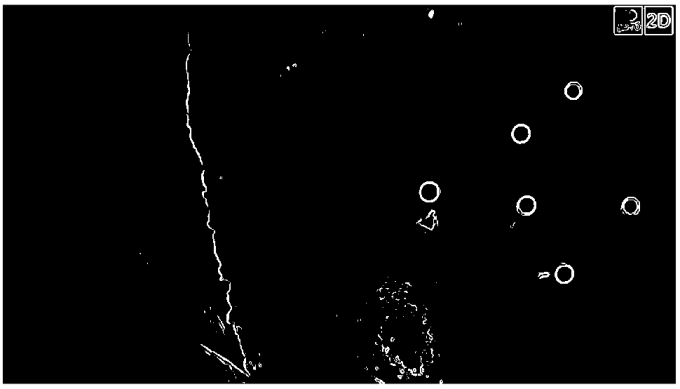


Figure 3.11: Detected edges using the Prewitt kernels with a threshold of 0.05. Similar to the Sobel kernels, the fiducials are sufficiently defined while the amount of noise is substantially smaller than for the previous threshold.

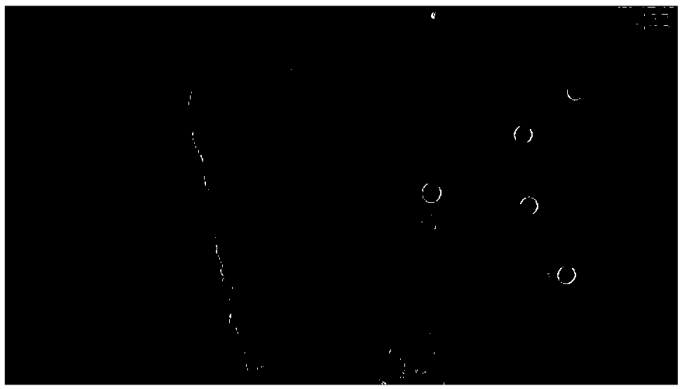


Figure 3.12: Detected edges using the Prewitt kernels with a threshold of 0.1. As before, a threshold 0f 0.1 generates a binary image infeasible for identifying the fiducials.

3.4.2 Filling hollow regions

In this part the hollow regions are filled in order to obtain the fiducials. The flood-fill algorithm in section 2.3.1 is applied for every image in the previous section.

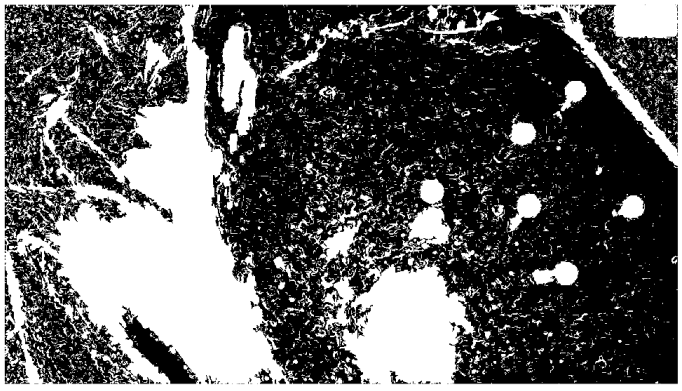


Figure 3.13: Reconstructed fiducials post edge detection with Sobel kernels and a threshold of 0.005. As predicted for this threshold, the noise affects the separability of the clustered regions of the fiducials.

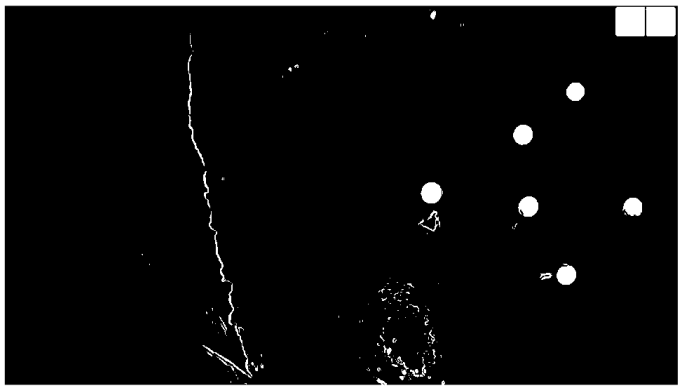


Figure 3.14: Reconstructed fiducials post edge detection with Sobel kernels and a threshold of 0.05. The reconstructed fiducials are feasibly highlighted and do not overlap with noise.

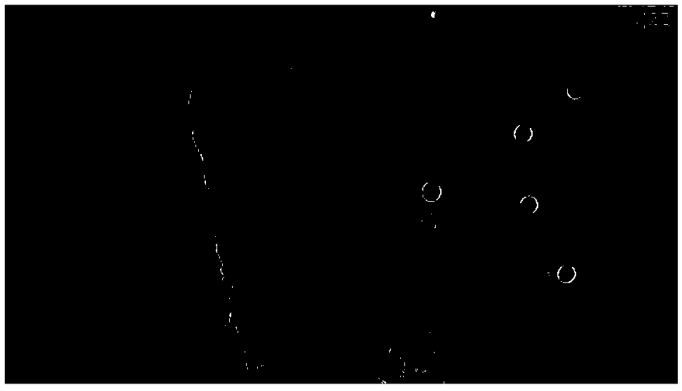


Figure 3.15: Reconstructed fiducials post edge detection with Sobel kernels and a threshold of 0.1. Since the fiducial edges found were incomplete, the morphological reconstruction was rendered ineffective.

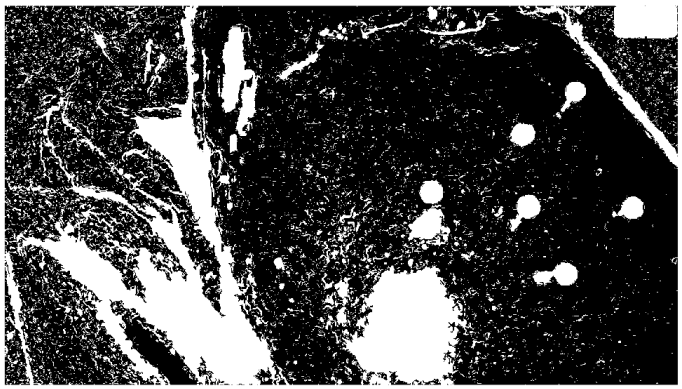


Figure 3.16: Reconstructed fiducials post edge detection with Roberts cross kernels and a threshold of 0.005. Similar to the Sobel counterpart, the excessive noise makes the fiducial recognition difficult.

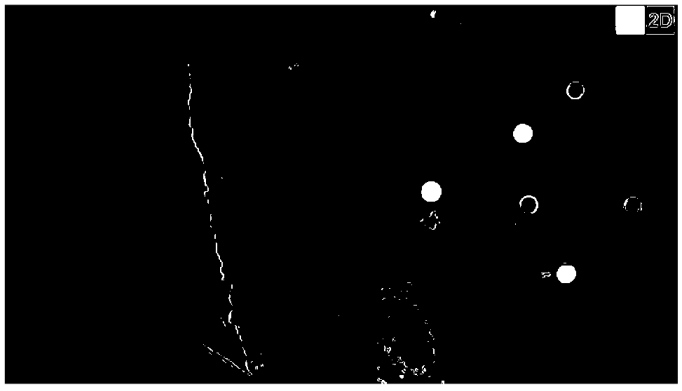


Figure 3.17: Reconstructed fiducials post edge detection with Roberts cross kernels and a threshold of 0.05. The reconstructed only captured three fiducials, as the edge detection left the remaining fiducials incomplete.

Figure 3.18: Reconstructed fiducials post edge detection with Roberts cross kernels and a threshold of 0.1. Since the edge detection was virtually ineffective, the reconstruction was impractical.

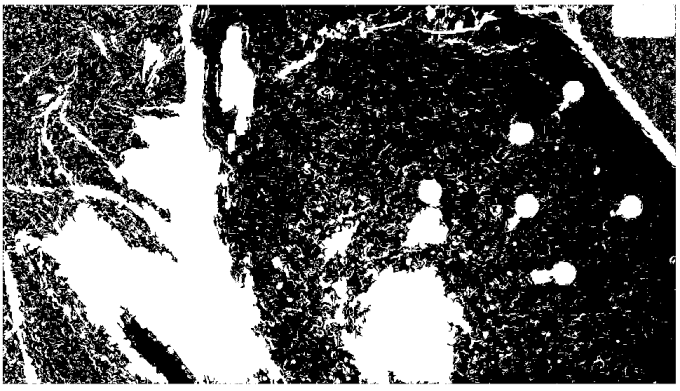


Figure 3.19: Reconstructed fiducials post edge detection with Prewitt kernels and a threshold of 0.005. The results are similar to the edge detection using Sobel kernels.

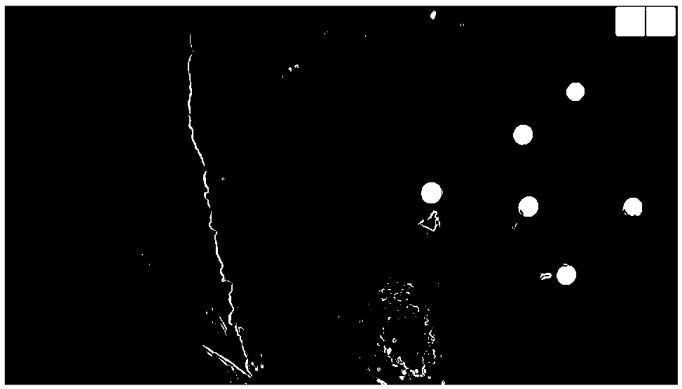


Figure 3.20: Reconstructed fiducials post edge detection with Prewitt kernels and a threshold of 0.05. Every fiducial are successfully captured, with minimal noise.

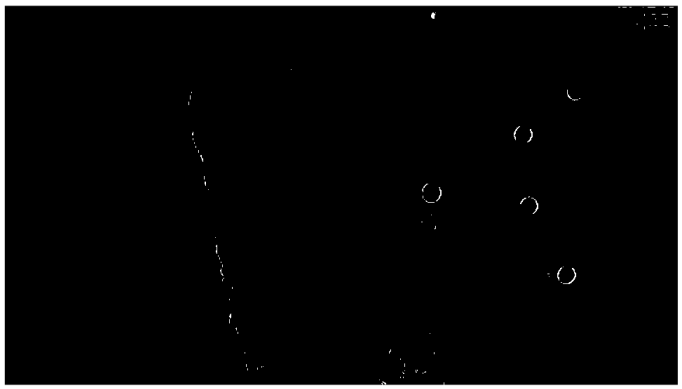


Figure 3.21: Reconstructed fiducials post edge detection with Prewitt kernels and a threshold of 0.1. The edges are not adequately defined in order to fill the hollow fiducials.

The most stable threshold seems to be 0.05 for every kernel, as indicated by how the fiducials are presented as clear clusters in the binary images, as seen in figures 3.14, 3.17 and 3.20. However, the Roberts cross kernel is unfavorable to the remaining two filters based on the results, as all fiducials are not properly captured.

3.5 Region measuring and noise filtering

Knowing that there are at most six fiducials present at any given time, the number of identifications needs to be limited to that number. Thus, to avoid labeling false fiducials, the regions in the binary image need to satisfy a certain size. Consider a connected-component labeled region $B^{(l)} \subseteq B$, where $B^{(l)}$ consequently has the

same dimensions as B but only contains a single region of B, as specified by

$$B_{ij}^{(l)} = \begin{cases} 1, \ L_{ij} = l; \ l = 1, ..., k, \ k > 1 \\ 0, \ \text{otherwise} \end{cases}$$

where L is the label matrix acquired from the connected-component labeling.

The area a of a labeled region is simply the number of white pixels

$$a = \sum_{i}^{1080} \sum_{j}^{1918} B_{ij}^{(k)}, \ k = 1, ..., l$$

The unwanted regions are removed if they are smaller than 1000 pixels and larger than 3000 pixels. As the perceived size in the laparoscopic view is not tracked in any way, and there is no general notion as to how the fiducial size varies in the 2D view, the limits set are based on experimental results. That is, the size range consistently included the fiducial whilst removing the majority of noise. The resulting binary image of the image segmentation after clearing false fiducials, based on size, is shown in figure 3.22.

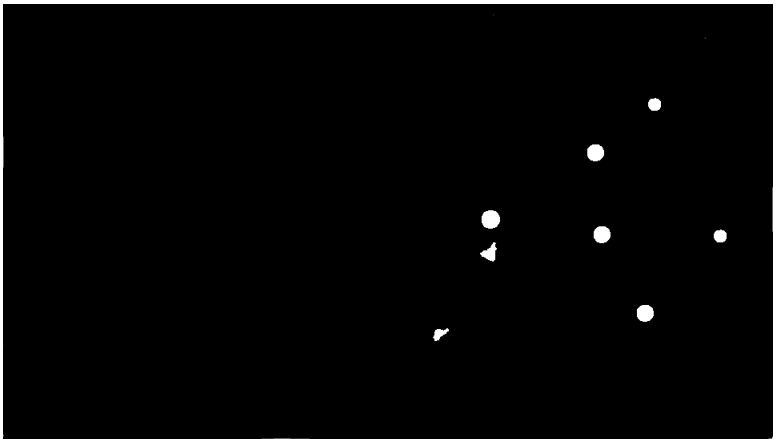


Figure 3.22: B after applying the size filter to the initial binary image in 3.3. In this particular case, most of the noise is discarded, leaving behind two regions which are similar in size to the fiducials.

Similarly, the binary images of the edge detection with threshold 0.05, where the noise regions smaller than 1000 pixels and larger than 3000 pixels are removed, are shown in figure 3.23, 3.24 and 3.25.

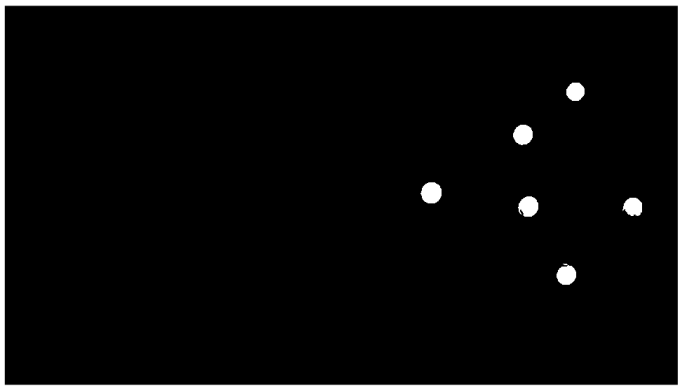


Figure 3.23: Post-filtering of the binary image of the Sobel edge detector in figure 3.14. The fiducials are well-shaped and no noise is present in this particular frame.

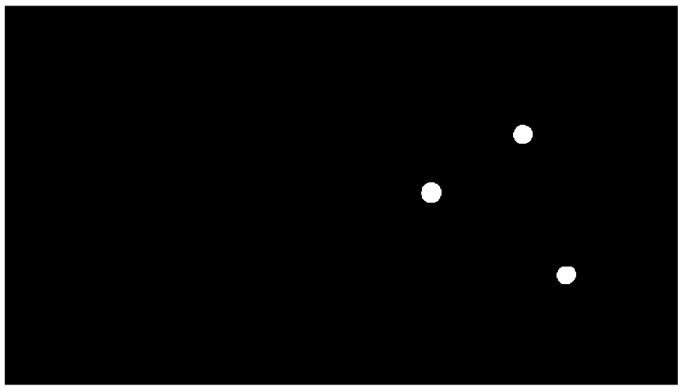


Figure 3.24: Post-filtering of the binary image of the Roberts cross edge detector in figure 3.17. Though the noise is removed, only three fiducials are captured.

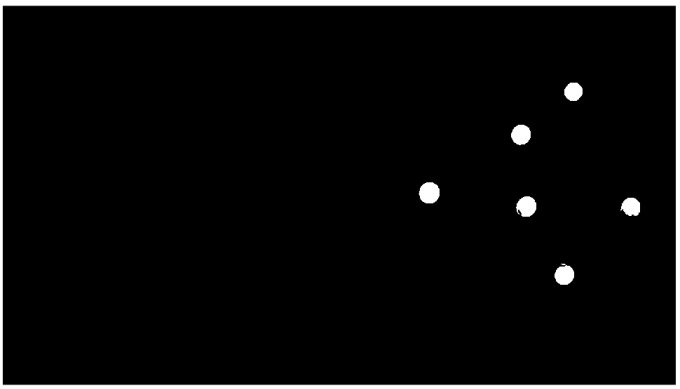


Figure 3.25: Post-filtering of the binary image of the Prewitt edge detector in figure 3.20. The results from the filter is almost identical to that of the Sobel edge detector in figure 3.23.

Comparing the edge detection strategy utilizing Sobel and Prewitt filters with the image segmentation strategy, the obvious difference is how some noise remains in the result of the latter. Between the methods, the fiducials themselves do not deviate substantially in terms of shape but the edge detected fiducials are slightly larger in size.

Once the fiducials exclusively remain in the post-filtered binary image, their centroids can be determined. To find the coordinates of a centroid, the row and column indices of the nonzero elements of $B^{(l)}$, $R^{(l)}$ and $C^{(l)}$ respectively, need to be determined. The sets of these indices are defined as

$$C^{(l)} := \{j \mid B^{(l)}_{\cdot j} = 1;\ j = 1, ..., 1918\}$$
$$R^{(l)} := \{i \mid B^{(l)}_{i\cdot} = 1;\ i = 1, ..., 1080\}$$

The centroid coordinates then equal to the mean of the elements in these sets. That is,

$$(\hat{x}_l,\, \hat{y}_l) = (\frac{\sum_{n=1}^{N} C_n^{(l)}}{N},\, \frac{\sum_{n=1}^{M} R_m^{(l)}}{M})$$

where N and M are the number of elements in sets $C^{(l)}$ and $R^{(l)}$, respectively.

3.6 Error analysis

Assuming that the fiducial detection is overall continuous and feasible, where fiducials are only missed sporadically and false fiducials[1] are scarce, an error analysis of the centroid positions is performed. Since the relevant measurements should account for the amount of deviation between the estimated and true values, euclidean

[1]Labeling a part of the liver or other surroundings as a a fiducial. This occurs usually when noise in the binary image are of the same size as the fiducials.

distance is used for estimating the error. In particular, the error for the centroid positions is given by the two-dimensional euclidean distance

$$d_{\text{centroid}} = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}$$

where (x, y) and $(\hat{x}, \hat{y})$ denote the true and the estimated centroid. Note, since the true centroids are not measured in real-time, they remain unknown until after examining the films. To then establish these values, user-annotation is implemented and the error estimation is statistically studied, as opposed to tracked and displayed alongside the video.

3.7 Optimization

As previously mentioned, a central stage of this experiment involves maintaining an uninterrupted identification of the fiducials. The steps required to reach such a solution are essentially covered when describing the method for background subtraction and noise-filtering. However, in order to make the solution practicable the difference between the runtimes of processed and unprocessed films need to be close to zero.

To this end, the approach to reducing runtime primarily involves reducing input size of the video frames before applying either the image segmentation or the edge detection, as profiling shows that this part constitutes the majority of the algorithm's computational time. The input size is minimized by confining the fiducials in the video frame to a smaller search area before continuing with either process. Seeing as there are periods of minor camera movement, this strategy is sensible. This rectangular area can be viewed can be viewed as a two-dimensional integer lattice $S \subseteq Z^2$ composed of the coordinates for the area of interest. The corners of the search area are decided based on the four outermost fiducials. Boxes containing each fiducial are defined in the binary image B. The margins of these boxes are at most 100 pixels starting from the centroid. Then, the margin between the fiducials and S is defined by the box margins of the outermost fiducials. To determine the search area, the horizontal and vertical coordinates of every centroid in the frame are sorted, with the corners of S being $(x^{(1)} - n_1, y^{(1)} - m_1)$, $(x^{(2)} + n_2, y^{(1)} - m_1)$, $(x^{(2)} + n_2, y^{(2)} + m_2)$, and $(x^{(1)} - n_1, y^{(2)} + m_2)$, where coordinates of subscript 1 are closest to the point (1, 1), which is the upper left corner of the frame, and subscript 2 refers to the furthest coordinates. The values n_1, m_1, n_2 and m_2 corresponds to the margin between the outer fiducials and the borders of S and is given by,

$$n_1 = \begin{cases} 100, \text{ if } x - 100 \geq 0 \\ -100, \text{ if } x - 100 < 0 \end{cases}$$

$$n_2 = \begin{cases} 100, \text{ if } x + 100 \leq 1918 \\ 100 - 1918, \text{ if } x + 100 > 1918 \end{cases}$$

$$m_1 = \begin{cases} 100, \text{ if } y - 100 \geq 0 \\ -100, \text{ if } y - 100 < 0 \end{cases}$$
$$m_2 = \begin{cases} 100, \text{ if } y + 50 \leq 1080 \\ 100 - 1080, \text{ if } y + 50 > 1080 \end{cases}$$

The confined area is illustrated in figure 3.26.

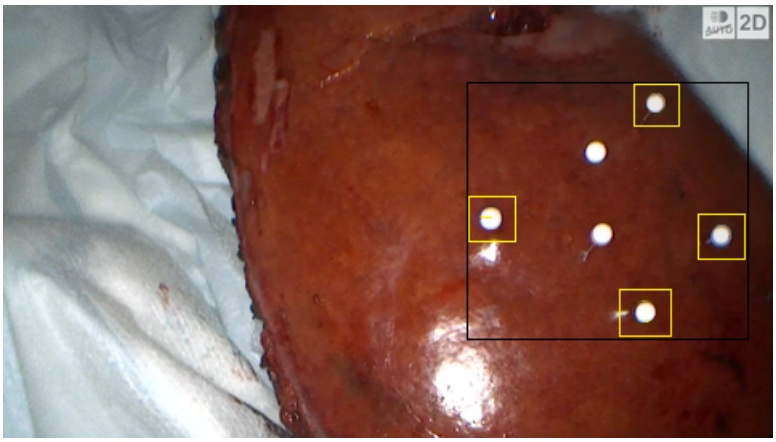


Figure 3.26: An illustration of how the search area S is established. The four outermost fiducials serve to set the sides of the S, whose perimeter is colored black. The space between these fiducials and the borders of S may be adjusted to ensure that it is small enough to minimize the input size for the image segmentation and edge detection, while maintaining infrequent updates of S.

Once a fiducial reaches the border of S, defined by a nonzero sum of the perimeter of S or

$$\sum_{i=1}^{k} s_{i1} + \sum_{j=1}^{l} s_{1j} + \sum_{i=1}^{k} s_{il} + \sum_{j=1}^{l} s_{kj} > 0, \ \forall s_{ij} \in S$$

where $k = (x^{(2)}+n_2)-(x^{(1)}-n_1)$ and $l = (y^{(2)}+m_2)-(y^{(1)}-m_1)$ are the dimensions of S, then the next frame is processed in its entirety in order to update the search area.

While the objective is to set the search area as small as feasibly possible, allowing for S to be somewhat larger in the form of greater margin values m ultimately limits the amount of updates required. As mentioned, movement of the laparoscope is expected even when settling at a certain viewpoint, and reducing the size of S to the point where frequent updates occur would defeat the purpose of reducing overall computation time.

4
Results

In this section, the performance of the two recognition algorithms will be evaluated through robustness and statistical means. The findings presented are derived from video frames obtained from a particular test film. The images shown in this section are from the same experiment as that of in figure in section 3.1.

4.1 Robustness

Before an error evaluation can be performed, the stability of the algorithm must be assessed. That is, performing an error estimation on an algorithm that experiences interruptions in the identification of fiducials between frames is in large futile. As for the two models tested - the edge detection model and the image segmentation model - only the latter fulfilled this stability requirement. The edge detection based recognition model was promising in the product itself as noise removal was easier while the clusters making up the fiducials in the binary image were generally regular in shape and size. The model, however, was significantly less universal than its image segmentation counterpart, requiring more parameter-tuning between frames to ensure that fiducials were not removed during the noise filtering. For this reason, the error estimation in this section is only based on the results from the image segmentation model.

4.2 Error estimation

For one of the test films, comprised of 1794 video frames, 26 were elected for the error analysis. Considering that periods with minimal movement was common in the films, the focus was to sample frames of various angles and distances. It is also worth noting that cases where either fiducials were unidentified or surroundings were instead labeled, due to noise caused by brightness in the ex vivo environment, are not among the sampled frames. The occurrence of these frames will be further discussed in the next chapter. Furthermore, the sampled frames feature all six of the fiducials together, as was the case almost throughout the course of the test films.

Because the individual fiducials are of interest and an average estimation per frame may be skewed by outliers, the observations in this study consists of the errors of every centroid for each of the 26 frames. That is, for 26 frames where every fiducial is present, 156 centroid errors are computed.

To showcase the result of the procedure, three frames are presented below in figures 4.1, 4.2 and 4.3, for short, intermediate, and long distances, as well as varying angle.

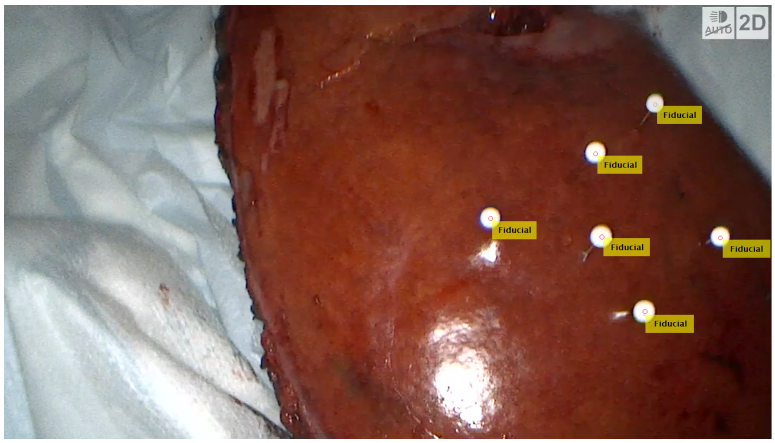


Figure 4.1: The fiducials captured at a proximity between 80 and 90 millimeter

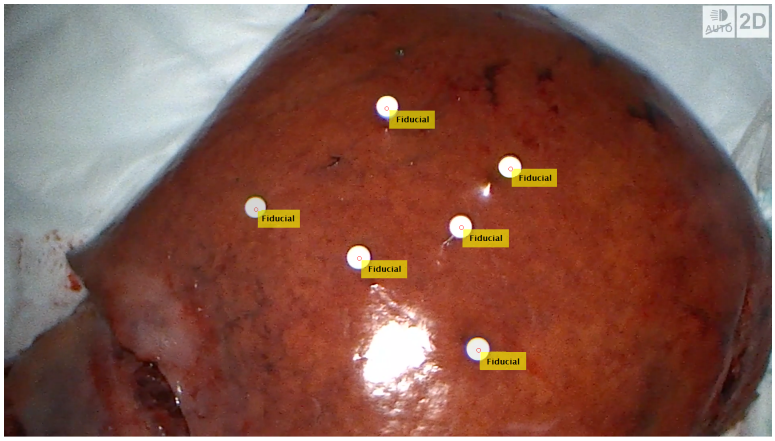


Figure 4.2: The fiducials captured at a proximity between 70 and 80 millimeter.

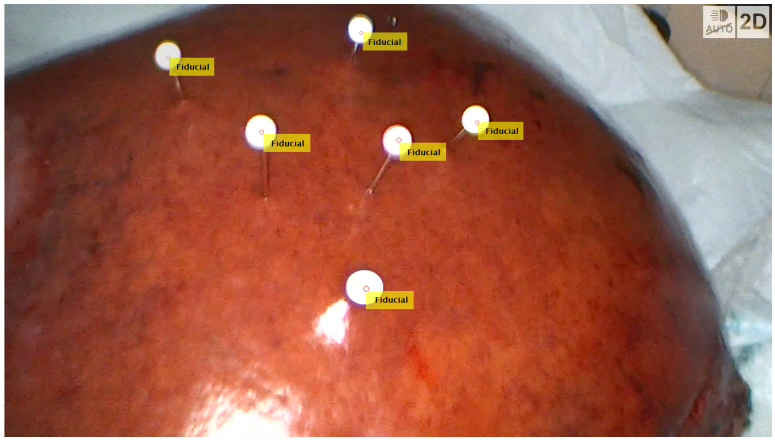


Figure 4.3: The fiducials captured at a proximity between 60 and 70 millimeter. Compared to the previous two frames, brightness-induced noise is more common which increases the number of disruptions to the fiducial recognition between frames.

The distribution of the errors are visualized in the histogram of figure 4.4. The sample mean and variance of the errors are approximately 2.94 and 2.35. Most of the observations fall in the range of 1 to 5 pixels in error, with a few around 10 pixels in error.

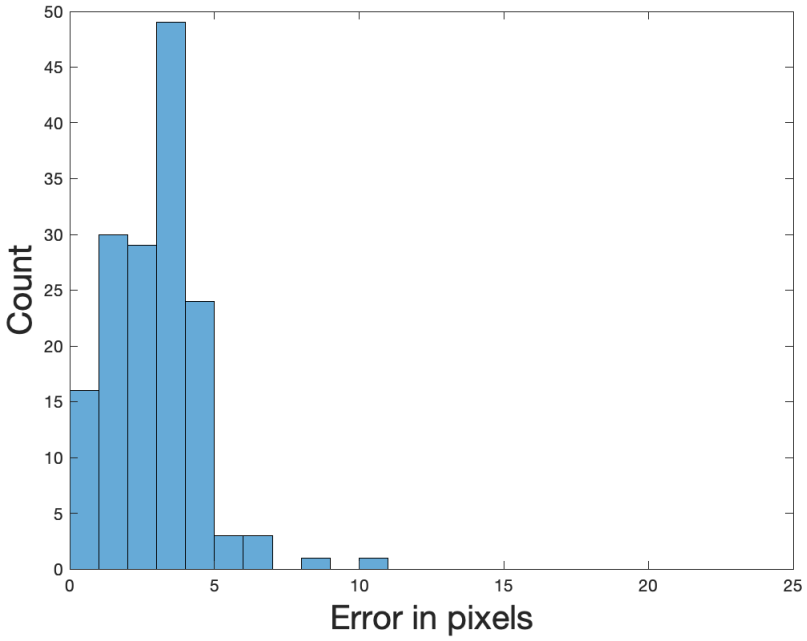


Figure 4.4: The distribution of centroid errors, totaling 156 observations, with a sample mean and variance of 2.94 and 2.35, respectively. Most of the errors are between 1 to 5 pixels which, divided by the pixel height of 1080, yields a percentage error of 0.09% to 0.46%.

5

Discussion

With a series of experiments conducted and a recognition model based on image segmentation constructed, a multitude of aspects regarding the performance of this model as well as the effectiveness of the overall premise can be raised. The points discussed in the following paragraphs include a review of the results presented and the performance of the method. Similarly, the second model implementing edge detection will be discussed in terms of its promises and failures. Moreover, other topics, such as the difficulties with respect to the translation from ex vivo to in vivo as well as other potential sources of noise not covered in this project, are discussed. To summarize this chapter, a list of items for possible future work will be introduced.

An object recognition model based brightness segmentation was constructed under the notion that the liver, being a fairly homogeneous organ, makes for a suitable background on which fiducials are easily identified in an image. In large, the results agrees with this assessment. Furthermore, accounting for the fact that the identification and centroid estimation of the fiducials was overall uninterrupted and uniform, the surgeon utilizing the laparoscope did not need to adjust the procedure to accommodate for the recognition model's functionality. Subsequently, the model's functionality and the surgeon's activity, the POSIT assumptions were largely fulfilled.

A second model based on edge detection was explored with varying success. While the model performed rather well in singular instances, it struggled to maintain a continuous identification between video frames. The issue primarily lies in need for fine-tuning the threshold frequently to capture every fiducial. Considering that some success was found in this method, there might be an incentive to continue future work in order to circumvent this obstacle.

The first recognition model is fairly robust in terms of constant detection of the fiducials and accuracy. The intermediate proximity, or a camera distance between 75 and 85 millimeters, was especially proven to be quite accurate. However, as the general notion is to run the algorithm throughout the course of an operation, the prevalence of certain distances is not fully grasped. For example, there was a shortage of frames with closer viewpoints, and considerable number of those remaining suffered from significant levels of brightness. Whether this is a persistent problem, or such shots during actual liver resection surgery are frequent, is therefore not fully understood.

As this experiment involved filming in an ex vivo environments with various backgrounds, there was considerable variability in the background colors captured. For instance, during the first few experiments a liver specimen was placed on a cork disk whose surface consisted of similar tones and shapes to some fiducials; a stark contrast to white paper towel used in the test films analyzed in the results. Further, the white plastic surface of the ex vivo environment affected the brightness as it reflected the camera light onto the glossy surfaces of the liver specimen and the fiducials, rendering the model somewhat ineffective at close proximities. Although the camera light intensity could be adjusted, working under lower light levels was not always a viable option for the surgeons operating. In a final experiment, where non-reflective paper was wrapped inside the ex vivo environment to cover the plastic surface, brightness was nevertheless still a problem in close proximities. Since the series of tests have yet been attempted during actual surgery, it is difficult to predict how compatible the settings will be between patients. To propose a solution, having an option to calibrate the segmentation threshold until the fiducials are sufficiently clustered may address this issue. Again, this goes by the assumption that the reflection of light will be manageable and similar between patients.

Other sources of noise that were not tested for were, for example, blood spreading on the fiducial heads during the course of the surgery. The experiments were controlled in a manner such that various noise factors could be reduced, among which were casing the ex vivo environment in non-reflective paper and removing stains from the fiducials. If staining is a common occurrence during surgery it might interfere with the recognition algorithm, which in turn may become a distracting task for the surgical team to maintain the fiducials without displacing them from their position on the liver. Another potential noise factor is caused by cirrhosis, where scarring is present on the liver surface. This is especially a problem for the edge detection model, as the edges of the scars may affect the fiducial recognition.

5.0.1 Future Work

As it stands now, a framework for fiducial recognition based on image segmentation is defined with clear areas of improvement. Finding ways to minimize noise levels is still a crucial point, and modifying the setup of the experiment is a viable option, as seen with the use of non-reflective paper to counter brightness. To further grasp the model's performance, conducting the fiducial recognition in vivo still remains.

Moreover, combining this algorithm with the POSIT remains to be examined. In this endeavor, the recognition algorithm would be applied on frames of a simulated liver with fiducials, whose locations are known beforehand. The positions identified will then be used as input to compute and generate AR corresponding to a tumor. A crucial aspect to inspect is the magnitude of output error in POSIT which may be generated due to erroneous input following the estimated 2D positions in the laparoscopic view.

6
Conclusion

The objective of this thesis was to expand on an established model for projecting AR in liver resection surgery by adding an algorithm for finding the fiducials in frames captured by the laparoscope. Moreover, by emulating a scenario of laparoscopic liver resection at Sahlgrenska University hospital, a greater understanding of the procedure and how to adapt the fiducial recognition model to accommodate the needs of the surgeons, as well as the assumptions of the POSIT algorithm, was achieved.

In all, proper selection of fiducials to facilitate the identification and to eliminate noise were adequately found. This resulted in a robust model where white fiducials was regularly identified through image segmentation and whose center positions, were estimated with success. A second model based on edge detection was explored with varying success, but lacked the same robustness as the former model.

An error analysis was perform to validate the recognition method. It was found that while the positions were generally off by less than about %0.9, brightness-induced noise caused by the laparoscopic camera light at certain proximities made detection ineffective, despite attempts to remedy the noise. Plans to address this problem were suggested for future work.

Lastly, a variety of techniques to achieve a real-time playback of the processed frames were explored. It was shown that the computation time comprised mostly of the image segmentation. By limiting this process to an area of the image containing the fiducials, a near real-time playback was achieved.

References

[1] Takeshi Takahara, Go Wakabayashi, Toru Beppu, Arihiro Aihara, Kiyoshi Hasegawa, Naoto Gotohda, Etsuro Hatano, Yoshinao Tanahashi, Toru Mizuguchi, Toshiya Kamiyama, Tetsuo Ikeda, Shogo Tanaka, Nobuhiko Taniai, Hideo Baba, Minoru Tanabe, Norihiro Kokudo, Masaru Konishi, Shinji Uemoto, Atsushi Sugioka, and Tadahiro Takada. Long-term and perioperative outcomes of laparoscopic versus open liver resection for hepatocellular carcinoma with propensity score matching: A multi-institutional japanese study. *Journal of hepato-biliary-pancreatic sciences*, 22, 06 2015. doi: 10.1002/jhbp.276.

[2] Rui Tang, Long-Fei Ma, Zhi-Xia Rong, Mo-Dan Li, Jian-Ping Zeng, Xue-Dong Wang, Hong-En Liao, and Jia-Hong Dong. Augmented reality technology for preoperative planning and intraoperative navigation during hepatobiliary surgery: A review of current methods. *Hepatobiliary Pancreatic Diseases International*, 17(2):101 – 112, 2018. ISSN 1499-3872. doi: https://doi.org/10.1016/j.hbpd.2018.02.002. URL `http://www.sciencedirect.com/science/article/pii/S149938721830047X`.

[3] Cancer. 9 2018. URL `https://www.who.int/news-room/fact-sheets/detail/cancer`. [Accessed 31 May 2020].

[4] Andrea Teatini, Egidijus Pelanis, Davit L Aghayan, Rahul Prasanna Kumar, Rafael Palomar, Åsmund Avdem Fretland, Bjørn Edwin, and Ole Jakob Elle. The effect of intraoperative imaging on surgical navigation for laparoscopic liver resection surgery. *Scientific Reports*, 9(1):1–11, 2019.

[5] Laparoscopic liver resection. URL `https://www.massgeneral.org/digestive/treatments-and-services/laparoscopic-liver-resection`. [Accessed 31 May 2020].

[6] Xavier Untereiner, Audrey Cagnet, Riccardo Memeo, Vito Blasi, Stylianos Tzedakis, Tullio Piardi, François Severac, Didier Mutter, Reza Kianmanesh, Jacques Marescaux, Daniele Sommacale, and Patrick Pessaux. Short-term and middle-term evaluation of laparoscopic hepatectomies compared with open hepatectomies: A propensity score matching analysis. *World Journal of Gastrointestinal Surgery*, 8:643, 09 2016. doi: 10.4240/wjgs.v8.i9.643.

[7] What is augmented reality? URL `https://www.fi.edu/what-is-augmented-reality`. [Accessed 1 June 2020].

[8] Kevin Phuong. Object recognition and tracking for augmented

reality, 5 2014. URL https://www.mathworks.com/videos/object-recognition-and-tracking-for-augmented-reality-90546.html. [Accessed 11 February 2020].

[9] Nicolas C. Buchs, Francesco Volonte, François Pugin, Christian Toso, Matteo Fusaglia, Kate Gavaghan, Pietro E. Majno, Matthias Peterhans, Stefan Weber, and Philippe Morel. Augmented environments for the targeting of hepatic lesions during image-guided robotic liver surgery. *Journal of Surgical Research*, 184(2):825 – 831, 2013. ISSN 0022-4804. doi: https://doi.org/10.1016/j.jss.2013.04.032. URL http://www.sciencedirect.com/science/article/pii/S0022480413004083.

[10] Lisa Månsson. A model for an augmented reality tool in tumour removal laparoscopic surgery. Master's thesis, Chalmers University of Technology, 2020.

[11] Rodrigo Verschae and Javier Ruiz-del Solar. Object detection: current and future directions. *Frontiers in Robotics and AI*, 2:29, 2015.

[12] Rubén Laguna, Rubén Barrientos, L. Felipe Blázquez, and Luis J. Miguel. Traffic sign recognition application based on image processing techniques. *IFAC Proceedings Volumes*, 47(3):104 – 109, 2014. ISSN 1474-6670. doi: https://doi.org/10.3182/20140824-6-ZA-1003.00693. URL http://www.sciencedirect.com/science/article/pii/S1474667016416009. 19th IFAC World Congress.

[13] Jianxin Wu, Adebola Osuntogun, Tanzeem Choudhury, Matthai Philipose, and James M Rehg. A scalable approach to activity recognition based on object use. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007.

[14] *Convert RGB image or colormap to grayscale*. URL https://www.mathworks.com/help/matlab/ref/rgb2gray.html. [Accessed 25 May 2020].

[15] *Understanding Color Spaces and Color Space Conversion*. URL https://www.mathworks.com/help/images/understanding-color-spaces-and-color-space-conversion.html. [Accessed 25 May 2020].

[16] *Find connected components in binary image*. URL https://www.mathworks.com/help/images/ref/bwconncomp.html. [Accessed 24 May 2020].

[17] Michael B Dillencourt, Hanan Samet, and Markku Tamminen. A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM (JACM)*, 39(2):253–280, 1992. doi: https://doi.org/10.1145/128749.128750.

[18] Steve Eddins. Connected component labeling – part 4, 4 2007. URL https://blogs.mathworks.com/steve/2007/04/15/connected-component-labeling-part-4/. [Accessed 19 May 2020].

[19] David Jacobs. Image gradients. 2005.

[20] *Edge detection methods for finding object boundaries in images*, . URL

https://www.mathworks.com/discovery/edge-detection.html. [Accessed 10 February 2020].

[21] *Find edges in intensity image*, . URL https://www.mathworks.com/help/images/ref/edge.html. [Accessed 10 February 2020].

[22] Ashley Walker Robert Fisher, Simon Perkins and Erik Wolfart. Sobel edge detector. .

[23] Ashley Walker Robert Fisher, Simon Perkins and Erik Wolfart. Roberts cross edge detector. .

[24] *Flood-Fill Operations*, . URL https://www.mathworks.com/help/images/flood-fill-operations.html. [Accessed 20 April 2020].

[25] *Fill image regions and holes*, . URL https://www.mathworks.com/help/images/ref/imfill.html. [Accessed 10 February 2020].